```python
# Required libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Extinction coefficients
k_v = 0.510
k_b = 0.757

# Visual magnitudes
heze_mag_v = [-16.83301237,-16.88772898,-16.90694106,-16.9166258,-
16.85617632,-16.76175901,-16.66974569,-16.67259044]
her_mag_v = [-14.83188811,-15.68023224,-15.83390523,-16.00939752,-
15.97681243,-16.03223959,-15.98663884,-16.01906571]
yed_mag_v = [-17.54996365,-17.81656384,-17.81835167,-17.60092995,-
17.91160608,-17.11816473,-17.89733586,-17.68017079]
rasalas_mag_v = [-16.84151275,-16.81815968]
tau_mag_v = [-16.0663,-16.06822172]

# Blue magnitudes
heze_mag_b = [-16.59560186,-16.7811887,-16.7692966,-16.7492507,-
16.73827997,-16.56101596,-16.04331042,-16.49638265]
her_mag_b = [-14.67390169,-15.22277176,-15.49079774,-15.60459509,-
15.6835147,-13.82930808,-15.62401559,-15.61371424]
yed_mag_b = [-16.70765196,-16.87975909,-16.98093434,-17.02958407,-
17.027593,-16.58595811,-16.59420416,-16.58789537]
rasalas_mag_b = [-16.0319594,-16.019593]
tau_mag_b = [-15.88377,-15.909128]

# Visual airmass
heze_secz_v = [1.190, 1.087, 1.061, 1.067, 1.118, 1.193, 1.268, 1.351]
her_secz_v = [3.738, 2.282, 1.765, 1.443, 1.222, 1.125, 1.077, 1.051]
yed_secz_v = [2.210, 1.697, 1.449, 1.275, 1.145, 1.107, 1.092, 1.089]
rasalas_secz_v = [1.696, 1.699]
tau_secz_v = [1.066, 1.068]

# Blue airmass
heze_secz_b = [1.191, 1.087, 1.061, 1.067, 1.117, 1.185, 1.266, 1.349]
her_secz_b = [3.767, 2.278, 1.770, 1.446, 1.224, 1.126, 1.078, 1.051]
yed_secz_b = [2.219, 1.702, 1.452, 1.277, 1.145, 1.107, 1.092, 1.089]
rasalas_secz_b = [1.696, 1.699]
tau_secz_b = [1.066, 1.068]

# Function to compute corrected magnitudes
def correct_magnitude(mag_list, secz_list, extinction_coeff):
    return [m - s * extinction_coeff for m, s in zip(mag_list,
secz_list)]

# Corrected V-band magnitudes
heze_v_corr = correct_magnitude(heze_mag_v, heze_secz_v, k_v)
```

```python
her_v_corr = correct_magnitude(her_mag_v, her_secz_v, k_v)
yed_v_corr = correct_magnitude(yed_mag_v, yed_secz_v, k_v)
rasalas_v_corr = correct_magnitude(rasalas_mag_v, rasalas_secz_v, k_v)
tau_v_corr = correct_magnitude(tau_mag_v, tau_secz_v, k_v)

# Corrected B-band magnitudes
heze_b_corr = correct_magnitude(heze_mag_b, heze_secz_b, k_b)
her_b_corr = correct_magnitude(her_mag_b, her_secz_b, k_b)
yed_b_corr = correct_magnitude(yed_mag_b, yed_secz_b, k_b)
rasalas_b_corr = correct_magnitude(rasalas_mag_b, rasalas_secz_b, k_b)
tau_b_corr = correct_magnitude(tau_mag_b, tau_secz_b, k_b)

# Optional: Calculate B-V color indices
heze_bv = [b - v for b, v in zip(heze_b_corr, heze_v_corr)]
her_bv = [b - v for b, v in zip(her_b_corr, her_v_corr)]
yed_bv = [b - v for b, v in zip(yed_b_corr, yed_v_corr)]
rasalas_bv = [b - v for b, v in zip(rasalas_b_corr, rasalas_v_corr)]
tau_bv = [b - v for b, v in zip(tau_b_corr, tau_v_corr)]

# Display Results
print("\nCorrected V magnitudes:")
print("Heze:", heze_v_corr)
print("Her:", her_v_corr)
print("Yed:", yed_v_corr)
print("Rasalas:", rasalas_v_corr)
print("Tau:", tau_v_corr)

print("\nCorrected B magnitudes:")
print("Heze:", heze_b_corr)
print("Her:", her_b_corr)
print("Yed:", yed_b_corr)
print("Rasalas:", rasalas_b_corr)
print("Tau:", tau_b_corr)

print("\nB-V Color Indices:")
print("Heze:", heze_bv)
print("Her:", her_bv)
print("Yed:", yed_bv)
print("Rasalas:", rasalas_bv)
print("Tau:", tau_bv)
```

```
Corrected V magnitudes:
Heze: [-17.43991237, -17.442098979999997, -17.44805106, -17.4607958, -
17.42635632, -17.370189009999997, -17.31642569, -17.36160044]
Her: [-16.73826811, -16.84405224, -16.73405523, -16.74532752, -
16.600032430000002, -16.60598959, -16.53590884, -16.55507571]
Yed: [-18.677063649999997, -18.68203384, -18.55734167, -18.25117995, -
18.49555608, -17.68273473, -18.45425586, -18.235560789999997]
Rasalas: [-17.70647275, -17.68464968]
```

```
Tau: [-16.609959999999997, -16.61290172]

Corrected B magnitudes:
Heze: [-17.497188859999998, -17.604047700000002, -17.572473600000002,
-17.5569697, -17.583848970000002, -17.458060959999997, -
17.001672420000002, -17.51757565]
Her: [-17.52552069, -16.94721776, -16.83068774, -16.69921709, -
16.6100827, -14.681690080000001, -16.44006159, -16.40932124]
Yed: [-18.38743496, -18.16817309, -18.08009834, -17.996273069999997, -
17.894358, -17.42395711, -17.420848160000002, -17.41226837]
Rasalas: [-17.3158314, -17.305736]
Tau: [-16.690732, -16.717604]

B-V Color Indices:
Heze: [-0.05727648999999957, -0.16194872000000515, -
0.12442254000000119, -0.09617390000000015, -0.1574926500000018, -
0.08787195000000025, 0.3147532699999971, -0.15597521000000114]
Her: [-0.7872525800000005, -0.10316552000000101, -0.09663250999999917,
0.046110429999998814, -0.010050269999997141, 1.924299509999999,
0.09584725000000205, 0.14575446999999997]
Yed: [0.28962868999999714, 0.5138607499999992, 0.4772433300000003,
0.2549068800000036, 0.6011980799999996, 0.25877762000000004,
1.033407699999998, 0.8232924199999978]
Rasalas: [0.3906413499999992, 0.37891368000000014]
Tau: [-0.08077200000000317, -0.10470228000000148]

import numpy as np

# Function to compute temperature from B−V
def compute_temperatures(bv_values):
    bv_values = np.array(bv_values)
    temperatures = 8540 / (bv_values + 0.865)
    return temperatures

# Function to compute average temperature
def compute_average_temperature(temperatures):
    return np.mean(temperatures)

# Compute temperatures for each observation
heze_temp = compute_temperatures(heze_bv)
her_temp = compute_temperatures(her_bv)
yed_temp = compute_temperatures(yed_bv)
rasalas_temp = compute_temperatures(rasalas_bv)
tau_temp = compute_temperatures(tau_bv)

# Compute average temperatures
heze_avg_temp = compute_average_temperature(heze_temp)
her_avg_temp = compute_average_temperature(her_temp)
yed_avg_temp = compute_average_temperature(yed_temp)
rasalas_avg_temp = compute_average_temperature(rasalas_temp)
```

```
tau_avg_temp = compute_average_temperature(tau_temp)

# Display all temperatures
print("All Temperatures (in K):")
print(f"Heze Temperatures: {heze_temp}")
print(f"Her Temperatures: {her_temp}")
print(f"Yed Temperatures: {yed_temp}")
print(f"Rasalas Temperatures: {rasalas_temp}")
print(f"Tau Temperatures: {tau_temp}")

# Display average temperatures
print("\nAverage Temperatures (in K):")
print(f"Heze    → Average Temp: {heze_avg_temp:.2f} K")
print(f"Her     → Average Temp: {her_avg_temp:.2f} K")
print(f"Yed     → Average Temp: {yed_avg_temp:.2f} K")
print(f"Rasalas → Average Temp: {rasalas_avg_temp:.2f} K")
print(f"Tau     → Average Temp: {tau_avg_temp:.2f} K")
```

```
All Temperatures (in K):
Heze Temperatures: [10572.92488614 12147.05135022 11531.54188625
11107.84350323
 12070.5459809  10989.17996847  7238.80171996 12044.7128513 ]
Her Temperatures: [109842.87324261  11209.78404653  11114.47336222
9373.17773873
   9988.89139365   3061.70060597   8887.98921993   8449.13404143]
Yed Temperatures: [7396.31716582 6193.51881617 6362.4827251
7625.634017   5824.5881757
 7599.36828071 4498.50682759 5058.36542226]
Rasalas Temperatures: [6801.30516568 6865.42815415]
Tau Temperatures: [10889.69024314 11232.44194393]

Average Temperatures (in K):
Heze    → Average Temp: 10962.83 K
Her     → Average Temp: 21491.00 K
Yed     → Average Temp: 6319.85 K
Rasalas → Average Temp: 6833.37 K
Tau     → Average Temp: 11061.07 K
```

```
#with considering error in extinction coeff

import numpy as np

# Flux values provided for each observation (B and V bands)
heze_flux_b = [4508701.6830509, 5170004.0507298, 5151795.78927873,
5058805.59655851, 4955659.10592348, 4250464.47954029,
2625816.77673634, 4006463.01677547]
heze_flux_v = [5410095.73124256, 5689729.14731996, 5791304.83731021,
5843194.17781927, 5526759.03191712, 5066448.17814183,
4654770.52050429, 4666982.55813034]
her_flux_b = [740560.72410642, 1227746.49341138, 1571517.05182064,
```

```
1745171.20003324, 1876747.63821216, 340191.32198417, 1776667.8138284,
1759890.69608209]
her_flux_v = [856554.97918934, 1871082.32384133, 2155569.80790761,
2533722.26622797, 2458810.00718146, 2587592.21585164,
2481164.37008575, 2556385.13779051]
yed_flux_b = [4820152.55911799, 5648116.36412053, 6199743.72886315,
6483860.000642, 6471980.51248597, 4309066.97070744, 4341918.46259516,
4316762.4112353]
yed_flux_v = [10470934.92079779, 13385220.7204699, 13407279.67108685,
10974177.47134178, 14609738.13885884, 7035028.9857678,
14418973.57572909, 11805063.18418531]
rasalas_flux_b = [2586890.226, 2557628.013]
rasalas_flux_v = [5452618.34, 5336590.417]
tau_flux_b = [2256877.568, 2310208.45]
tau_flux_v = [2670053.725, 2674783.823]


# Given extinction coefficient errors
extinction_error_b = 0.178 # Error in extinction coefficient (B band)
extinction_error_v = 0.333  # Error in extinction coefficient (V band)

# Function to compute uncertainty in magnitudes from flux values
def compute_magnitude_uncertainty(flux_values):
    return (2.5 / np.log(10)) * (1 / np.sqrt(flux_values))



# Updated function to compute the color index uncertainty
def compute_color_index_uncertainty(delta_m_b, delta_m_v):
    # Compute color index uncertainty with the additional terms for
extinction coefficients
    return np.sqrt(delta_m_b**2 + delta_m_v**2 + extinction_error_b**2
+ extinction_error_v**2)


# Function to compute uncertainty in temperature from color index
uncertainty
def compute_temperature_uncertainty(bv_values, delta_bv):
    dT_dBV = -8540 / (bv_values + 0.865)**2
    return np.abs(dT_dBV * delta_bv)

# Compute uncertainties for magnitudes (B and V bands)
heze_delta_m_b = [compute_magnitude_uncertainty(f) for f in
heze_flux_b]
heze_delta_m_v = [compute_magnitude_uncertainty(f) for f in
heze_flux_v]
her_delta_m_b = [compute_magnitude_uncertainty(f) for f in her_flux_b]
her_delta_m_v = [compute_magnitude_uncertainty(f) for f in her_flux_v]
yed_delta_m_b = [compute_magnitude_uncertainty(f) for f in yed_flux_b]
yed_delta_m_v = [compute_magnitude_uncertainty(f) for f in yed_flux_v]
```

```
rasalas_delta_m_b = [compute_magnitude_uncertainty(f) for f in
rasalas_flux_b]
rasalas_delta_m_v = [compute_magnitude_uncertainty(f) for f in
rasalas_flux_v]
tau_delta_m_b = [compute_magnitude_uncertainty(f) for f in tau_flux_b]
tau_delta_m_v = [compute_magnitude_uncertainty(f) for f in tau_flux_v]


# Example usage for each dataset:
heze_delta_bv = [compute_color_index_uncertainty(db, dv)
                 for db, dv in zip(heze_delta_m_b, heze_delta_m_v)]
her_delta_bv = [compute_color_index_uncertainty(db, dv)
                for db, dv in zip(her_delta_m_b, her_delta_m_v)]
yed_delta_bv = [compute_color_index_uncertainty(db, dv)
                for db, dv in zip(yed_delta_m_b, yed_delta_m_v)]
rasalas_delta_bv = [compute_color_index_uncertainty(db, dv)
                   for db, dv in zip(rasalas_delta_m_b,
rasalas_delta_m_v)]
tau_delta_bv = [compute_color_index_uncertainty(db, dv)
                for db, dv in zip(tau_delta_m_b, tau_delta_m_v)]





# Given the calculated temperatures (as per your previous data)
heze_bv = [0.330, 0.274, 0.331, 0.390, 0.291, 0.392, 0.671, 0.298]  #
Example B-V values for Heze
her_bv = [1.108, 0.915, 0.930, 1.072, 1.029, 1.478, 1.079, 1.095]  #
Example B-V values for Her
yed_bv = [0.504, 0.508, 0.521, 0.393, 0.456, 0.330, 0.567, 0.537]  #
Example B-V values for Yed
rasalas_bv = [0.789, 0.820]  # Example B-V values for Rasalas
tau_bv = [0.419, 0.439]  # Example B-V values for Tau

# Compute temperature uncertainties (sigma_T) for each observation
heze_temp_uncertainty = [compute_temperature_uncertainty(bv, delta_bv)
for bv, delta_bv in zip(heze_bv, heze_delta_bv)]
her_temp_uncertainty = [compute_temperature_uncertainty(bv, delta_bv)
for bv, delta_bv in zip(her_bv, her_delta_bv)]
yed_temp_uncertainty = [compute_temperature_uncertainty(bv, delta_bv)
for bv, delta_bv in zip(yed_bv, yed_delta_bv)]
rasalas_temp_uncertainty = [compute_temperature_uncertainty(bv,
delta_bv) for bv, delta_bv in zip(rasalas_bv, rasalas_delta_bv)]
tau_temp_uncertainty = [compute_temperature_uncertainty(bv, delta_bv)
for bv, delta_bv in zip(tau_bv, tau_delta_bv)]

# Display the uncertainties for color index and temperature
print("Uncertainty in Color Index (B-V):")
print(f"Heze Color Index Uncertainty: {heze_delta_bv}")
```

```python
print(f"Her Color Index Uncertainty: {her_delta_bv}")
print(f"Yed Color Index Uncertainty: {yed_delta_bv}")
print(f"Rasalas Color Index Uncertainty: {rasalas_delta_bv}")
print(f"Tau Color Index Uncertainty: {tau_delta_bv}")

# Display uncertainties in temperature
print("\nUncertainty in Temperature (in K):")
print(f"Heze Temperature Uncertainty: {heze_temp_uncertainty}")
print(f"Her Temperature Uncertainty: {her_temp_uncertainty}")
print(f"Yed Temperature Uncertainty: {yed_temp_uncertainty}")
print(f"Rasalas Temperature Uncertainty: {rasalas_temp_uncertainty}")
print(f"Tau Temperature Uncertainty: {tau_temp_uncertainty}")
```

```
Uncertainty in Color Index (B-V):
Heze Color Index Uncertainty: [np.float64(0.37758903499481966),
np.float64(0.37758897652927487), np.float64(0.3775889727844573),
np.float64(0.3775889759605369), np.float64(0.3775889976785235),
np.float64(0.3775890755998127), np.float64(0.37758933007476847),
np.float64(0.3775891243378523)]
Her Color Index Uncertainty: [np.float64(0.377592330477677),
np.float64(0.37759050593721766), np.float64(0.37759011770835),
np.float64(0.3775899107901837), np.float64(0.3775898668509316),
np.float64(0.37759359203310233), np.float64(0.3775899079836354),
np.float64(0.37758989784734803)]
Yed Color Index Uncertainty: [np.float64(0.37758887317010575),
np.float64(0.37758879323921535), np.float64(0.3775887684568046),
np.float64(0.377588783237462), np.float64(0.3775887482831707),
np.float64(0.3775889843902316), np.float64(0.3775888680207723),
np.float64(0.37758889408693286)]
Rasalas Color Index Uncertainty: [np.float64(0.377589289950271),
np.float64(0.37758930307844024)]
Tau Color Index Uncertainty: [np.float64(0.37758967653087355),
np.float64(0.37758965953020834)]

Uncertainty in Temperature (in K):
Heze Temperature Uncertainty: [np.float64(2258.0909709954376),
np.float64(2485.5913529188283), np.float64(2254.316106348968),
np.float64(2047.3388388774695), np.float64(2413.023401430921),
np.float64(2040.8295601100979), np.float64(1366.769103511608),
np.float64(2384.064045416728)]
Her Temperature Uncertainty: [np.float64(828.3747731422766),
np.float64(1017.7448935436936), np.float64(1000.8052716007197),
np.float64(859.4468232941606), np.float64(898.9142233482704),
np.float64(587.4053652542619), np.float64(853.2685286214221),
np.float64(839.3944522116702)]
Yed Temperature Uncertainty: [np.float64(1720.5613481833757),
np.float64(1710.550468568941), np.float64(1678.612596080945),
np.float64(2037.5847099061557), np.float64(1847.8694256113627),
np.float64(2258.090668365454), np.float64(1572.5013132087577),
```

```
np.float64(1640.5182099255019)]
Rasalas Temperature Uncertainty: [np.float64(1178.708804632979),
np.float64(1135.7369170424604)]
Tau Temperature Uncertainty: [np.float64(1955.9058030138854),
np.float64(1896.3687076503509)]
```

```
import numpy as np
import pandas as pd

# Provided temperatures and uncertainties for each star
heze_temp = [8600.41071789, 8609.37722339, 8585.0619761,
8585.69992079,
             8567.34124688, 8547.01074766, 8487.26610773,
8533.97558657]
her_temp = [4334.25814277, 5310.55340168, 5722.30894659,
6362.35340455,
            6395.69428303, 6927.27413472, 6400.03110673,
6379.20141911]
yed_temp = [5582.13279203, 6225.86675412, 6341.28969012, 6424.7131623,
            6526.30203515, 6086.0810548, 6179.33106669, 6076.04685379]
rasalas_temp = [8592.36018703, 8592.36018703]
tau_temp = [8584.78197365, 8585.06098743]

heze_temp_uncertainty = [2258.0909709954376, 2485.5913529188283,
2254.316106348968,
                         2047.3388388774695, 2413.023401430921,
2040.8295601100979,
                         1366.769103511608, 2384.064045416728]
her_temp_uncertainty = [828.3747731422766, 1017.7448935436936,
1000.8052716007197,
                        859.4468232941606, 898.9142233482704,
587.4053652542619,
                        853.2685286214221, 839.3944522116702]
yed_temp_uncertainty = [1720.5613481833757, 1710.550468568941,
1678.612596080945,
                        2037.5847099061557, 1847.8694256113627,
2258.090668365454,
                        1572.5013132087577, 1640.5182099255019]
rasalas_temp_uncertainty = [1178.708804632979, 1135.7369170424604]
tau_temp_uncertainty = [1955.9058030138854, 1896.3687076503509]

# Combine into single lists for the DataFrame
data = {
    "Star": ["Heze"] * len(heze_temp) + ["Hercules"] * len(her_temp) +
            ["Yed Posterior"] * len(yed_temp) + ["Rasalas"] *
len(rasalas_temp) +
            ["Tau Virginis"] * len(tau_temp),
    "Temperature (K)": heze_temp + her_temp + yed_temp + rasalas_temp
+ tau_temp,
    "Temperature Uncertainty (K)": heze_temp_uncertainty +
```

```
her_temp_uncertainty +
                                   yed_temp_uncertainty +
rasalas_temp_uncertainty + tau_temp_uncertainty
}

# Create DataFrame
df = pd.DataFrame(data)

# Display the table
print("Temperature Data:\n")
print(df)

# Actual known temperatures
actual_temps = {
    "Heze": 6288,
    "Hercules": 4124,
    "Yed Posterior": 4918,
    "Rasalas": 4519,
    "Tau Virginis": 8413
}

# Calculate average temps and RMS uncertainties
print("\nAverage Temperatures and RMS Errors (with RMS Uncertainty):")
stars = {
    "Heze": (heze_temp, heze_temp_uncertainty),
    "Hercules": (her_temp, her_temp_uncertainty),
    "Yed Posterior": (yed_temp, yed_temp_uncertainty),
    "Rasalas": (rasalas_temp, rasalas_temp_uncertainty),
    "Tau Virginis": (tau_temp, tau_temp_uncertainty)
}

for star, (temps, uncerts) in stars.items():
    avg_temp = np.mean(temps)
    rms_uncertainty = np.sqrt(np.mean(np.square(uncerts)))
    actual_temp = actual_temps[star]


    print(f"{star}:")
    print(f"  Avg Temp = {avg_temp:.2f} K")
    print(f"  RMS Uncertainty = ±{rms_uncertainty:.2f} K")
    print(f"  Actual Temp = {actual_temp} K")
```

```
Temperature Data:

             Star  Temperature (K)  Temperature Uncertainty (K)
0            Heze      8600.410718                  2258.090971
1            Heze      8609.377223                  2485.591353
2            Heze      8585.061976                  2254.316106
3            Heze      8585.699921                  2047.338839
```

```
4            Heze      8567.341247                      2413.023401
5            Heze      8547.010748                      2040.829560
6            Heze      8487.266108                      1366.769104
7            Heze      8533.975587                      2384.064045
8        Hercules      4334.258143                       828.374773
9        Hercules      5310.553402                      1017.744894
10       Hercules      5722.308947                      1000.805272
11       Hercules      6362.353405                       859.446823
12       Hercules      6395.694283                       898.914223
13       Hercules      6927.274135                       587.405365
14       Hercules      6400.031107                       853.268529
15       Hercules      6379.201419                       839.394452
16  Yed Posterior      5582.132792                      1720.561348
17  Yed Posterior      6225.866754                      1710.550469
18  Yed Posterior      6341.289690                      1678.612596
19  Yed Posterior      6424.713162                      2037.584710
20  Yed Posterior      6526.302035                      1847.869426
21  Yed Posterior      6086.081055                      2258.090668
22  Yed Posterior      6179.331067                      1572.501313
23  Yed Posterior      6076.046854                      1640.518210
24        Rasalas      8592.360187                      1178.708805
25        Rasalas      8592.360187                      1135.736917
26   Tau Virginis      8584.781974                      1955.905803
27   Tau Virginis      8585.060987                      1896.368708

Average Temperatures and RMS Errors (with RMS Uncertainty):
Heze:
  Avg Temp = 8564.52 K
  RMS Uncertainty = ±2182.03 K
  Actual Temp = 6288 K
Hercules:
  Avg Temp = 5978.96 K
  RMS Uncertainty = ±869.48 K
  Actual Temp = 4124 K
Yed Posterior:
  Avg Temp = 6180.22 K
  RMS Uncertainty = ±1821.18 K
  Actual Temp = 4918 K
Rasalas:
  Avg Temp = 8592.36 K
  RMS Uncertainty = ±1157.42 K
  Actual Temp = 4519 K
Tau Virginis:
  Avg Temp = 8584.92 K
  RMS Uncertainty = ±1926.37 K
  Actual Temp = 8413 K
```